

REMARKS

In response to the Office Action mailed on May 2, 2007, Applicant(s) respectfully request(s) reconsideration.

Claim(s) 1-51 are now pending in this Application.

In this Amendment, claim(s) 1, 20-21, 23, 29, 32-34, 49-51 have been amended and claim(s) 18-19 have been canceled.

Claim(s) 1, 29, 32-34 and 49-51 are independent claims and the remaining claims are dependent claims. Applicant(s) believe that the claim(s) as presented are in condition for allowance. A notice to this affect is respectfully requested.

Rejections under 35 U.S.C. §112:

The Office Action rejects the use of the term "significant." Claim usage of this term has been amended to recite "reportable" events, as disclosed at page 3, lines 10-14.

Claim 20 has been amended to remove the trade specific references alluded to by the rejection.

Accordingly, it is respectfully requested that the rejection under 35 U.S.C. §112 be withdrawn.

Rejections under 35 U.S.C. §101:

Claim 50 has been rejected as non-statutory subject matter. Claim 50 has have been herein amended to clarify that they are directed to a computer readable storage medium operable to store computer program logic embodied in computer program code as a set of computer instructions encoded thereon, and are therefore now believed within the statutory breadth of 35 U.S.C. §101. It is therefore respectfully requested that the rejection under 35 U.S.C. §101 be withdrawn.

Rejections under 35 U.S.C. §102(b) based on Silberschatz, et al:

The Office Action rejects claims 1, 14, 15, 18, 19, 21-23, 34 and 49-51 under 35 U.S.C. §102(b) based on Silberschatz, et al. Applicants respectfully disagree with these contentions and assert that the present claimed invention is not anticipated by any disclosure in the Silberschatz reference.

Referring specifically to the rejection of Claim 1, the disclosed system solves the problem of a disabled event handler at the time the event occurs by enabling the corresponding event handler (Fig. 3, 62) upon occurrence of the event, as recited in claim 1. Thus, the claimed invention enables and disables subscribers 60 to an event in response to the particular event occurring, rather than operating system (OS) activation for general processor scheduling as in Silberschatz. In further clarification, claim 1 has been amended with the subject matter of claims 18 and 19, to recite that enabling modules correspond[s] to activation of a corresponding component by an activation mechanism, disabling corresponding to deactivation of the corresponding component by the activation mechanism, the activation and deactivation operations operable to reduce memory consumption by inactive components and provide selective invocation to maintain availability of the component, and that enabling and disabling [is] performed at a level of granularity of the modules, each of the modules corresponding to a component and operable be enabled and disabled by activation and deactivation of the corresponding component as discussed in further detail at page 26, line 25-page 27, line 8 of the specification as filed.

The Office Action suggests that Silberschatz teaches the enabling and disabling of claims 18 and 19 at 6.1.4 and at 9.2. In contrast, however, the cited Silberschatz reference refers to general OS scheduling for all processes executed under a particular CPU. The Silberschatz reference differs because Silberschatz discloses context switching (activation/deactivation) based on time slicing, not event occurrences (Sec. 9.2, p. 264). As is known in the art, each processing unit in a multiprogramming system maintains its own Program Counter (PC) indicative of the next instruction for execution, independent of other programming units. The OS allocates the CPU to each of the PCs of the

programming units according to a scheduling mechanism (such as time slicing), which is typically time based.

Amended claim 1 is therefore submitted as allowable, because Silberschatz has no showing, teaching or disclosure of the claimed enabling and disabling by the activation mechanism, as now recited in amended claim 1. Amended claim 1 is now distinguishable because the cited reference refers to operating system scheduling, and generally concerns process scheduling that is outside the control of user applications. In other words, a user application typically cannot modify its own scheduling in order to obtain a larger slice of CPU time. Only privileged and generally inaccessible processes are able to modify the scheduling parameters relevant to the process control block (PCB, 4.1.3) and the dispatch responsive execution (6.1.4) of main CPU time.

Further, claim 49, similar in scope to claim 1 and rejected on similar grounds, has been herein amended with the subject matter of claims 18 and 19, to further clarify applicant's claimed invention. Since claim 49 is similar in scope to claim 1, it is further submitted that no new search issues are presented by amended claim 49 because the scope of the subject matter would have been encompassed by former claims 18 and 19.

Rejections under 35 U.S.C. §103(a) based on Silberschatz in view of Frank, U.S. Publication No. 2004/0250254:

The Office Action rejects claims 2-13, 16, 17, 20, 24, 25, 28-33 and 35-48 under 35 U.S.C. §103(a) based on Silberschatz in view of Frank, U.S. Publication No. 2004/0250254. Specifically, the Office Action also rejects claims 34, 50, and 51 on similar grounds as claim 1. Further to the problem solved by the present application, specifically dispatching an indication of an event to an enabled handler operable to process the event, the present invention further differs from Silberschatz and Frank '254 by employing a local event table and a persistent event table such that events receivable by disabled handlers are not lost pending

enablement of the subscriber (handler) for which the event is directed, discussed further at page 23, lines 15-19 and at page 26, lines 14-20.

The cited prior art makes no showing, teaching or disclosure of a local event table and a persistent event table for maintaining events regardless of the enablement state of the handler directed to process the event. Accordingly, claim 34, similar in scope to claim, 1 has been herein amended with the subject matter of claim 6 and 7 (dependent from claim 1) to recite that traversing further include[es]: identifying the event in a persistent event mapping, the persistent event mapping indicative of modules containing event handlers associated with the event; and dispatching, in the identified modules, the associated event handlers, dispatching comprising: selectively receiving an enablement indication in response to traversing in the persistent event mapping; identifying, in a local event mapping for the enabled module, subscriber entities including handlers associated with the mapped event; and invoking, in the mapped module, the identified subscribers including associated handlers, to further clarify and distinguish. The Office Action suggests that Frank '254, in conjunction with Silberschatz, discloses the claimed local and persistent event tables (Frank, [0099]). However, Frank differs because Frank simply teaches process enablement and disablement according to OS prescribed intervals, as discussed above with respect to claims 18 and 19. Frank does not show, teach or disclose, alone or in combination, the claimed persistent event mapping for maintaining an event in the event that the prescribed handler is disabled at the triggering of the event (page 4, line 29-page 5, line 6). The claimed invention solves this problem by persistent event mapping as now recited in amended claim 34. Further, claims 50 and 51 have been likewise amended, and further to recite that the association between the event identity and the corresponding event handler further compris[es] a local mapping in a local event map identifying a subscribing entity including an event handler corresponding to the event identity, and a persistent event mapping in a shared memory portion identifying the module

including the event handler corresponding to the event identity, as recited in claim 47 by way of further distinction.

Thus, the cited Frank '254 reference discloses virtual processing units for correlating threads with the processing units such that the virtual processor identifies the thread for receiving an event. Frank differs from the claimed invention because Frank makes no showing, teaching, or disclosure, alone or in combination, of event handling through process activation and deactivation (i.e. how to deliver an event to an inactive, or swapped out, process). In contrast, the present invention claims, as recited in claims 6 and 7, a local event mapping and a persistent event mapping for performing event dispatch regardless of the activation state of the recipient handler (subscriber), as disclosed at page 26, lines 12-20.

Claims 29 and 33, also rejected on Silberschatz in conjunction with Frank '254, have been further amended with subject matter of claims 14, 44 and 18, to recite that the local subscriber issuing the received event subscription becom[es] disabled following the subscription until an occurrence and subsequent publication of the event, invoking including indexing, in the persistent mapping via the event identity, a persistent reference to the modules including the event handlers associated with the event, the persistent reference operable to identify a handler independently of enablement of the module containing the associated event handler, in which enabling modules correspond[s] to activation of a corresponding component by an activation mechanism; and that disabling correspond[s] to deactivation of the corresponding component by the activation mechanism, the activation and deactivation operations operable to reduce memory consumption by inactive components and provide selective invocation to maintain availability of the component, as discussed above, to further clarify and distinguish the system defined by the present claims. Thus, in a system defined by the combination of Silberschatz and Frank '254, the problem of handler invocation for an event triggers while the corresponding handler/subscriber is

disabled (i.e. swapped out as in Silberschatz) is still not addressed. The claimed local and persistent event tables solve this problem. Claim 32 has been likewise amended.

As the remaining claims depend from, either directly or indirectly, from claims 1, 29 and 34, it is respectfully submitted that all claims are now in condition for allowance.

Applicant(s) hereby petition(s) for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-3735.

-27-

If the enclosed papers or fees are considered incomplete, the Patent Office is respectfully requested to contact the undersigned collect at (508) 616-9660, in Westborough, Massachusetts.

Respectfully submitted,

/CJL/

Christopher J. Lutz, Esq.
Attorney for Applicant(s)
Registration No.: 44,883
Chapin Intellectual Property Law, LLC
Westborough Office Park
1700 West Park Drive
Westborough, Massachusetts 01581
Telephone: (508) 616-9660
Facsimile: (508) 616-9661

Attorney Docket No.: EMC03-22(03111)

Dated: August 2, 2007